NATIONAL POLAR-ORBITING OPERATIONAL ENVIRONMENTAL SATELLITE SYSTEM (NPOESS)

OPERATIONAL ALGORITHM DESCRIPTION DOCUMENT FOR VIIRS SNOW COVER EDR SOFTWARE (D39592 Rev B)

CDRL No. A032

Northrop Grumman Space & Mission Systems Corporation
One Space Park
Redondo Beach, California 90278

NATIONAL POLAR-ORBITING OPERATIONAL ENVIRONMENTAL SATELLITE SYSTEM (NPOESS)

OPERATIONAL ALGORITHM DESCRIPTION DOCUMENT FOR VIIRS SNOW COVER EDR SOFTWARE (D39592 Rev B)

ELECTRONIC APPROVAL SIGNATURES:

____

Roy Tsugawa Date
Algorithm & Data Processing IPT Lead &
Algorithm Change Control Board Chairperson

____

Ben James Date
Operations & Support IPT Lead

The following individuals are recognized for their contributions to the current or previous versions of this document.

Robert Mahoney
Greg Richardson

Revision/Change Record	**For Document No.** D39592

Revision	Document Date	Revision/Change Description	Pages Affected
--	29 Apr2005	Initial PCIM Release - Reference ECR A048	All
A	4 May 2005	Revision A – Replace ITAR marking with Commerce Destination Control Statement. ECR A-196	Title Page
B	1 March 2010	In preparation for Public Release of this OAD, the following administrative changes were made: all ITAR markings were removed, and Distribution Statement F added. Approved for Public Release per Contracts Letter 100610-02.	All

LIST OF OAD TBD/TBR'S

No.	DESCRIPTION	PAGES AFFECTED
TBD01		

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

1.1 Objective

The purpose of this document is to describe the operational software method of solution for the core mathematical algorithms that produce the end-user data products. The National Polar-Orbiting Operational Environmental Satellite System (NPOESS) data products are individually known as Raw Data Records (RDRs), Temperature Data Records (TDRs), Sensor Data Records (SDRs) and Environmental Data Records (EDRs).

This document describes the software design of the operational algorithm as implemented in the IDPS and FTS and enables a software engineer to implement the algorithm without having specific science domain knowledge. Algorithmic changes incorporated due to infrastructure and timeliness requirements are reflected in this document.

1.2 Scope

Nominally, there is at least one Operational Algorithm Document (OAD) per EDR package. The scope of this document is limited to the description of the operational implementation of the algorithm(s) required to create the VIIRS Snow Cover EDR software.

1.3 Reference Documents

The primary software detailed design documents listed here, along with the latest listing of the "science grade" software are provided as attachments to this OAD.

1.3.1 Software Detailed Design Documents

Document Title	Document Number	Publication Date	Version
VIIRS Snow Cover Unit Detailed Design Document	Y3234	21 March 2005	V5, R5
VIIRS Snow/Ice Module Interface Control Document	Y0011650	21 March 2005	V5, R5
VIIRS Snow/Ice Module Software Architecture Document	Y2477	21 March 2005	V5, R6
VIIRS Snow/Ice Module Data Dictionary	Y2482	21 March 2005	V5, R5
VIIRS Snow Cover EDR Software Unit Test Report	NP-EMD.2005.510.0047	20 April 2005	Initial Release

1.3.2 Other Related Program Documents

Document Location/Title	Document Number	Publication Date	Version
VIIRS Snow Cover Algorithm Theoretical Basis Document	Y2401	22 March 2005	V5, R3
VIIRS Science Algorithms 3.3 Delivery to IDPS Package Version Description	D40174	20 April 2005	Initial Release

1.3.3 Source Code and Test Data References

Document Title	Publication Date	Version
$SDPSHOME/src/snowice/snow_cov/…	20 April 2005	ISTN_VIIRS_NGST_3.3
$SDPSHOME/TestCases/…	20 April 2005	ISTN_VIIRS_NGST_3.3

2. Operational Algorithm Description

The purpose of the Snow Cover EDR algorithm is to produce the following products:

- Snow Cover Binary Map (@ 375M resolution)
- Snow Fraction (@ 750M resolution).

This algorithm is heavily reliant on lookup tables which are all described in section 2.2.1. The overall processing chain is shown in Figure 1.

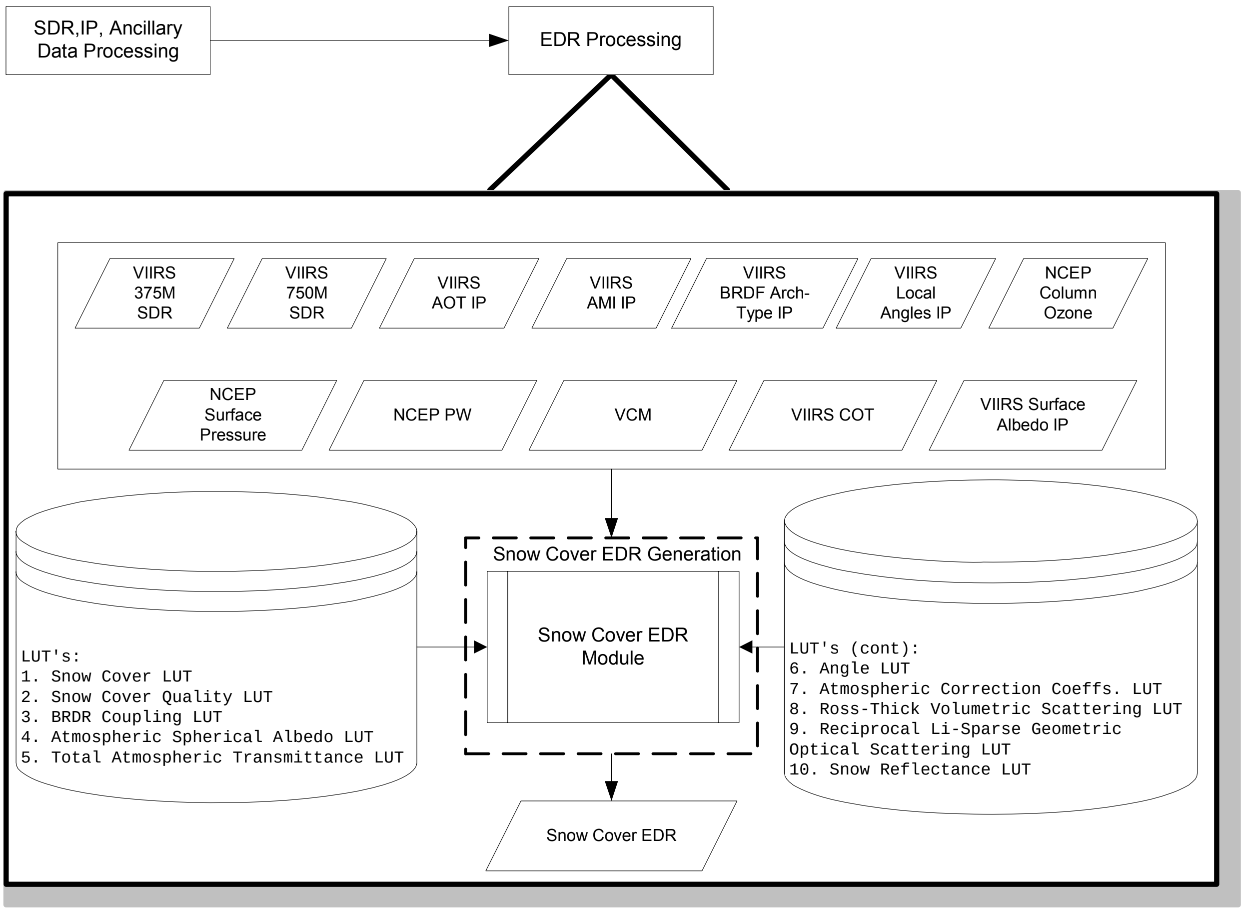

Figure 1: Snow Cover EDR Processing Chain

2.1 Science Support References

The Snow Cover EDR retrieval algorithm and the theoretical basis are described in detail in the VIIRS Snow Cover ATBD (Y2401).

2.1.1 Algorithm Theoretical Basis

The ATBD gives the theoretical basis for the Snow Binary Map algorithm in section 3.3.3; Snow Fraction computations are described in section 3.3.4. Further theoretical descriptions of non-snow and snow TOA reflectances are described in sections 3.3.4.1 and 3.3.4.2 respectively.

2.1.1.1 ATBD - Operational Adaptation, Deviation or Limitations

2.1.2 Prototype Code

The SM software module is written in Fortran 90, and has been compiled on an SGI Octane with 8GB of memory (OS is IRIX 6.5.15m) using f90 MIPSpro Compiler, version 7.3.1.3m.

2.1.2.1 Prototype Code - Adaptation, Deviation or Limitations

The Snow Cover/Depth EDR Algorithm is delivered with two options for computing snow cover fraction: 1) fraction computed by aggregation of the binary snow map 2) fraction computed by multiple end member spectral mixture method. NGST recommends that IDPS does not convert the spectral mixture method portion of the algorithm for operational and to leave the code intact pending further evaluation of the spectral mixture method.

The method of snow fraction computation is selectable based on the switch parameter “frac_option” defined in the snow_cover.lut run parameter configuration file. Currently, the frac_option option parameter is set to a value equal to 1 which directs the program to utilize the binary map aggregation method for snow fraction computation. It should be noted that NGST has commented out branches in the code that allow invoking the spectral mixture method in the delivered code so that the spectral mixture method may be implemented at a later time if deemed necessary.

2.2 Interfaces

2.2.1 Inputs

Input variable type sizes are specified in terms of bits. Variables highlighted in red are associated with the spectral mixture method snow fraction computation.

Table 1. Global Attributes (Ice Quality)

Input	Type	Description/Source	Units/Range
IBASCANS	int*32	Number of imagery resolution pixels in scan direction (VIIRS scan dimension)	Unitless/ IBASCANS > 0 ***(Currently set to 2708)***
IBATRACKS	int*32	Number of imagery resolution along-track lines in a VIIRS granule (VIIRS along-track dimension)	Unitless/ IBATRACKS > 0 ***(Currently set to 4032)***
MBASCANS	int*32	Number of moderate resolution pixels in scan direction (VIIRS scan dimension)	Unitless/ IBASCANS > 0 ***(Currently set to 1354)***
MBATRACKS	int*32	Number of moderate resolution along-track lines in a VIIRS granule (VIIRS along-track dimension)	Unitless/ IBATRACKS > 0 ***(Currently set to 2016)***
nbands_m	int*32	Number of Moderate Resolution Bands	Unitless/ nbands_m > 0 ***(Currently set to 9)***
nbands_i	int*32	Number of Imagery Resolution Bands	Unitless/ nbands_i > 0 ***(Currently set to 3)***
ntypes	int*32	Number of Snow Types 24 snow types = 6 grain sizes (50, 100, 250, 500, 750, 1000 micron radius) x 4 impurity levels (0, 1, 10, 100ppmw)	Unitless/ 1 ≤ ntypes ≤ 24 ***(Currently set to 24)***

Input	Type	Description/Source	Units/Range
n_ami	int*32	Number of Aerosol Models	Unitless/ 1 ≤ n_ami ≤ 9 ***(Currently using 2 models, "Continental" and "Marine", see*** Table 4***)***

Table 2. Main Inputs

Input	Data Type/Size	Description/Source	Units/Range
Reflectance_Mod	float*32 x MBASCANS x MBATRACKS x nbands_m	TOA Reflectances for all Moderate Resolution Bands	Unitless/ 0.0 ≥ Reflectance_Mod *FILL_VALUE = 65535* (Integer Scaled)
Reflectance_Img	float*32 x IBASCANS x IBATRACKS	TOA Reflectances for all Imagery Resolution Bands	Unitless/ 0.0 ≥ Reflectance_Img *FILL_VALUE = 65535* (Integer Scaled)
BrightTemp_M15	float*32 x MBASCANS x MBATRACKS	Band M15 Brightness Temperatures	Kelvin/ 0.0 ≥ bt15_m *FILL_VALUE = 65535* (Integer Scaled)
BrightTemp_M16	float*32 x MBASCANS x MBATRACKS	Band M16 Brightness Temperatures	Kelvin/ 0.0 ≥ bt16_m *FILL_VALUE = 65535* (Integer Scaled)
BrightTemp_I5	float*32 x IBASCANS x IBATRACKS	Band I5 Brightness Temperatures	Kelvin/ 0.0 ≥ bt5_i *FILL_VALUE = 65535* (Integer Scaled)
Latitude_Img	float*32 x IBASCANS x IBATRACKS	Latitude @ Imagery Resolution	Degrees/ -90 ≤ Latitude_Img ≤ 90 *FILL_VALUE = -999.9*
Longitude_Img	float*32 x IBASCANS x IBATRACKS	Longitude @ Imagery Resolution	Degrees/ -180 ≤ Longitude_Img ≤ 180 *FILL_VALUE = -999.9*
Latitude_Mod	float*32 x MBASCANS x MBATRACKS	Latitude @ Moderate Resolution	Degrees/ -90 ≤ Latitude_Mod ≤ 90 *FILL_VALUE = -999.9*
Longitude_Mod	float*32 x IBASCANS x IBATRACKS	Longitude @ Moderate Resolution	Degrees/ -180 ≤ Longitude_Mod ≤ 180 *FILL_VALUE = -999.9*
SolZenAng_Img	float*32 x IBASCANS x IBATRACKS	Solar Zenith Angle @ Imagery Resolution	Degrees/ -90 ≤ SolZenAng_Img ≤ 90 *FILL_VALUE = 32767* (Integer Scaled)
SenZenAng_Img	float*32 x IBASCANS x IBATRACKS	Sensor Zenith Angle @ Imagery Resolution	Degrees/ -90 ≤ SenZenAng_Img ≤ 90 *FILL_VALUE = 32767* (Integer Scaled)
SolAziAng_Img	float*32 x IBASCANS x IBATRACKS	Solar Azimuth Angle @ Imagery Resolution	Degrees/ -180 ≤ SolAziAng_Img ≤ 180 *FILL_VALUE = 32767* (Integer Scaled)
SenAziAng_Img	float*32 x IBASCANS x IBATRACKS	Sensor Azimuth Angle @ Imagery Resolution	Degrees/ -180 ≤ SenAziAng_Img ≤ 180 *FILL_VALUE = 32767* (Integer Scaled)

Input	Data Type/Size	Description/Source	Units/Range
SolZenAng_Mod	float*32 x IBASCANS x IBATRACKS	Solar Zenith Angle @ Moderate Resolution	Degrees/ -90 ≤ SolZenAng_Mod ≤ 90 *FILL_VALUE = 32767* (Integer Scaled)
SenZenAng_Mod	float*32 x IBASCANS x IBATRACKS	Sensor Zenith Angle @ Moderate Resolution	Degrees/ -90 ≤ SenZenAng_Mod ≤ 90 *FILL_VALUE = 32767* (Integer Scaled)
SolAziAng_Mod	float*32 x IBASCANS x IBATRACKS	Solar Azimuth Angle @ Moderate Resolution	Degrees/ -180 ≤ SolAziAng_Mod ≤ 180 *FILL_VALUE = 32767* (Integer Scaled)
SenAziAng_Mod	floa*t32 x IBASCANS x IBATRACKS	Sensor Azimuth Angle @ Moderate Resolution	Degrees/ -180 ≤ SenAziAng_Mod ≤ 180 *FILL_VALUE = 32767* (Integer Scaled)
AOT	float*32 x IBASCANSx IBATRACKS	Aerosol Optical Thickness (band M5) @ 750m	Unitless/ 0.0 ≤ AOT ≤ 1.0 *FILL_VALUE = -999.9*
VCM	See Table 3	See Table 3	See Table 3
AMI	uint*8 x MBASCANS x MBATRACKS	Aerosol Model Information/ VIIRS AMI IP	Unitless/ 0 to 9 (See Table 4); *FILL_VALUE = 255*
BRDF_Arch_Type	uint*16 x 3 x MBASCANS x MBATRACKS	The Update date gives when the last update of the Surface Albedo IP occurred for this cell. It is a Julian date from the year 2000 (i.e. 0 = 01/01/2000).	Day of year >0.0 FILL_VALULE = 32767
LVZA	float*32 x MBSCANS x MBTRACKS	Local View Zenith Angle (Local Angles IP)	Degrees/ -90 ≤ LVZA ≤ 90 *FILL_VALUE = 0*
LSZA	float*32 x MBSCANS x MBTRACKS	Local Solar Zenith Angle (Local Angles IP)	Degrees/ -90 ≤ LSZA ≤ 90 *FILL_VALUE = 0*
COT	float*32 x IBASCANS/2 x IBATRACKS/2	Cloud Optical Thickness @ 750m resolution	Unitless/ COT ≥ 0.0 *FILL_VALUE = 65535* (Integer Scaled)
O3	float*32 x MBASCANS x MBATRACKS	NCEP Column Ozone Ancillary Data	Atm-cm / 0.5 ≤ O3 ≤ 6.5 *FILL_VALUE = -999.00*
PW	float*32 x MBASCANS x MBATRACKS	NCEP Precipitable Water Ancillary Data	cm/ 0.0 ≤ PW ≤ 10.0 *FILL_VALUES = -999.00*
Surf_Pres	float*32 x MBASCANS x MBATRACKS	NCEP Surface Pressure Ancillary Data	millibars (mb)/ 200 ≤ Surf_Pres ≤ 1200 *FILL_VALUES = -999.00*
Snow_LUT	See Table 5	See Table 5	See Table 5

Input	Data Type/Size	Description/Source	Units/Range
Snow_Qual_LUT	See Note: fract_opt **parameter should be set to 1 for generating snow fraction based on 2x2 aggregation of the binary snow map.** Table 6	See Note: fract_opt **parameter should be set to 1 for generating snow fraction based on 2x2 aggregation of the binary snow map.** Table 6	See Note: fract_opt **parameter should be set to 1 for generating snow fraction based on 2x2 aggregation of the binary snow map.** Table 6
res_LUT	**See** Table 7	**See** Table 7	**See** Table 7
aero_LUT	**See** Table 8	**See** Table 8	**See** Table 8
trans_LUT	See Table 9	See Table 9	See Table 9
angle_LUT	See Table 10	See Table 10	See Table 10
robar_k1_LUT	See Table 11	See Table 11	See Table 11
robar_k2_LUT	See Table 12	See Table 12	See Table 12
Atm_Coeff_LUT	See Table 13	See Table 13	See Table 13
hk_ross_thick	See Table 14	See Table 14	See Table 14
rec_li_sparse_LUT	See Note: Code currently reads only the first value from the LUT table file Table 15	See Note: Code currently reads only the first value from the LUT table file Table 15	See Note: Code currently reads only the first value from the LUT table file Table 15
Snow_Refl_LUT	See Note: Code currently reads only the first value from the LUT table file Table 16	See Note: Code currently reads only the first value from the LUT table file Table 16	See Note: Code currently reads only the first value from the LUT table file Table 16

Table 3. VCM Bits Description

BYTE	Bit	Flag Description Key	Result			
0	0-1	Cloud Mask Quality	Bit 1	Bit 0		
			0	0		Poor
			0	1		Low
			1	0		Medium
			1	1		High
	2-3	Cloud Confidence	Bit 3	Bit 2		
			0	0		Confident Clear
			0	1		Probably Clear
			1	0		Probably Cloudy
			1	1		Confident Cloudy
	4	Day/Night	0 = Night 1 = Day			
	5	Snow/Ice Surface	0 = No 1 = Yes			
	6-7	Sun Glint	Bit 7	Bit 6		
			0	0		None
			0	1		Geometric Based
			1	0		Wind Speed Based
			1	1		Geometric and Wind Speed Based
1	0-2	Land/Water Background	Bit 2	Bit 1	Bit 0	
			0	0	0	Land & Desert
			0	0	1	Land, No Desert
			0	1	0	Inland Water
			0	1	1	Sea Water (Oceans)
			1	0	1	Coastal
	3	Shadow Detected	0 = No 1 = Yes			
	4	Non-Cloud Obstruction	0 = No 1 = Yes			
	5	Fire Detected	0 = No 1 = Yes			
	6	Thin Cirrus (solar – Band M9)	0 = No 1 = Yes			
	7	Thin Cirrus (IR – BTD)	0 = No 1 = Yes			
2	0	IR Threshold (BT M15)	1 = Cloud, 0 = No Cloud			
	1	High Cloud (M12 – M16)	1 = Cloud, 0 = No Cloud			
	2	Tri-Spectral IR Test	1 = Cloud, 0 = No Cloud			
	3	BTD Test (M15 – M12)	1 = Cloud, 0 = No Cloud			
	4	BTD Test (M12 – M13)	1 = Cloud, 0 = No Cloud			
	5	Visible Reflectance Test (M5)	1 = Cloud, 0 = No Cloud			
	6	Visible Reflectance Test (M7)	1 = Cloud, 0 = No Cloud			
	7	Visible Ratio (M7/M5)	1 = Cloud, 0 = No Cloud			
3	0-1	Adjacent Pixel Cloud Confident Value	Bit 1	Bit 0		
			0	0		Confident Clear
			0	1		Probably Clear
			1	0		Probably Cloudy
			1	1		Confident Cloudy

BYTE	Bit	Flag Description Key	Result				
	2	Conifer Boreal Forest	0 = No 1 = Yes				
	3	Spatial Uniformity Test	0 = No 1 = Yes				
	4-7	Imagery Resolution BTD Test (I5-I4). Each bit represents an imagery pixel, within a single moderate pixel.	Bit 7	Bit 6	Bit 5	Bit 4	
			0	0	0	0	None Cloudy
			1	1	1	1	All Cloudy
4	0-3	Imagery Resolution Test (I1)	Bit 7	Bit 6	Bit 5	Bit 4	
			0	0	0	0	None Cloudy
			1	1	1	1	All Cloudy
	4-7	Imagery Resolution Test (I2)	Bit 7	Bit 6	Bit 5	Bit 4	
			0	0	0	0	None Cloudy
			1	1	1	1	All Cloudy
5	0-2	Cloud Phase	Bit 2	Bit 1	Bit 0		
			0	0	0	Not Executed	
			0	0	1	Clear	
			0	1	0	Partly Cloudy	
			0	1	1	Water	
			1	0	0	Mixed Phase	
			1	0	1	Opaque Ice	
			1	1	0	Cirrus	
			1	1	1	Cloud Overlap	
	3-7	Spare	N/A				

Table 4. Aerosol Model Index IP

Aerosol Model	AMI Indices
Continental	0
Marine	1
Dust	2
Ocean	3
Smoke-High Absorption	4
Smoke-Low Absorption	5
Urban-High Absorption	6
Urban-Low Absorption	7
Dynamic Ocean Models	9
Fill Value	255

Table 5. Snow Cover LUT

Input	Data Type/Size	Description/Source	Units/Range
nbands_m	int*32	Number of moderate resolution bands	Unitless/ nbands > 0 ***(Currently set to 9)***
band_m	int*32 x nbands_m	Band Numbers (nbands_m in size)	Unitless/ `[1,2,3,4,5,7,8,10,11]`
num_r_water	int*32	Number of water reflectance thresholds (For I1 and I2)	Unitless/ num_r_water > 0 ***(Currently set to 2)***
r_water	float*32 x num_r_water	Water Reflectance Thresholds (For I1 and I2)	Unitless/ 0.0 ≤ r_water ≤ 1.0 r_water = `[0.10, 0.11]`
ndsi_thre1	float*32	First NDSI Threshold	Unitless/ ***(Currently set to 0.4)***
ndsi_thre2	float*32	Second NDSI Threshold	Unitless ***(Currently set to 0.1)***
n_max_coeff	int*32	Number of NDVI Maximum Coefficients	Unitless n_max_coeff > 0 ***(Currently set to 4)***
ndvi_max_coeff	float*32 x n_max_coeff	NDVI Maximum Coefficients	Unitless ndvi_max_coeff = `[-0.28,6.4,-12.0,10.0]`
n_min_coeff	int*32	Number of NDVI Minimum Coefficients	Unitless n_min_coeff > 0 ***(Currently set to 2)***
ndvi_min_coeff	float*32 x n_min_coeff	NDVI Minimum Coefficients	Unitless/ ndvi_min_coeff = `[0.32,-0.70]`
btmax	float*32	Brightness Temperature Threshold	Kelvin/ ***(Currently set to 283.0)***
ntypes	int*32	Number of Snow Types (6 grain size * 4 impurities = 24 types)	Unitless/ 1 ≤ ntypes ≤ 24 ***(Currently set to 24)***
frac_option	int*32	Flag which determines which snow fraction algorithm to run.	Untiless/ 0 = Spectral Mixing Algorithm **1 = Binary Snow Map Aggregation** 2 = Both **(Currently set to 1)**

Note: fract_opt parameter should be set to 1 for generating snow fraction based on 2x2 aggregation of the binary snow map.

Table 6. Snow Cover Quality LUT

Input	Data Type/Size	Description/Source	Units/Range
nbands_i	int*32	Number of Imagery Resolution Bands	Unitless/ nbands_i > 0 ***(Currently set to 3)***
nbands_m	int*32	Number of Moderate Resolution Bands	Unitless/ nbands_m > 0 ***(Currently set to 9)***
band_wgt	float*32 x nbands_m	Default Moderate Resolution Band Weights	Unitless/ 0.0 ≤ band_wgt ≤ 1.0 band_wgt = `[1.0,1.0,1.0, 1.0,1.0,1.0, 1.0,1.0,1.0]`
num_aot_bins	int*32	Number of AOT bins, which correspond to the number of AOT values used for thresholding (aot_bin, this table)	Unitless/ num_aot_bins > 0 ***(Currently set to 4)***
aot_bins	float*32 x num_aot_bins	AOT Bin Boundary Values	Unitless/ 0.0 ≤ aot_bin ≤ 1.0 aot_bin = `[0.0,0.15,0.5,1.0]`
num_thresh	int*32	Number of Solar Zenith Angle Thresholds	Unitless num_thresh > 0 ***(Currently set to 2)***
q_aot_sza	float*32 x (nbands_i + nbands_m) x num_aot_bins x num_thresh	Solar Zenith Angle values that correspond to the Solar Zenith Angle quality regimes (G/Y = "Green/Yellow", Y/R = "Yellow/Red", this corresponds to the "2" in the "Data Types/Size" column) and to the "aot_bin" values (**`(1) -> 0.0,(2) -> 0.15,(3) -> 0.5,(4) -> 1.0`**). The order for each num_aot_bin x num_thresh matrix of angles is: I1, I2, I3, M1, M2, M3, M4, M5, M7, M8, M10, and M11	Degrees/ -90 ≤ sza_bin ≤ 90 (**`I1,I2,I3,M1,M2,M3,M4, M5,M7,M8,M10,M11`**) **`(G/Y) (Y/R)`** `[75.0, 85.0;`**`(1)`** `70.0, 85.0;`**`(2)`** `65.0, 80.0;`**`(3)`** `60.0, 75.0]`**`(4)`** (Order of bands follows the scheme showed above)
cot_switch	int*32	Switch to flag the availability of the Cloud Optical Thickness IP	Unitless/ 0 = COT Not Available (Use VCM) 1 = COT Available (Use COT) ***(Currently set to 0)***
num_cloud_types	int*32	Number of Cloud Types	Unitless/ num_cloud_types > 0 *(**Currently set to 7**)*

Input	Data Type/Size	Description/Source	Units/Range
cloud_wgts	float*32 x num_cloud_types x nbands	Cloud weights corresponding to the 3 imagery bands + 9 moderate bands and the 7 cloud properties - 4 phases = Default (1), Water (2), Ice (3), Mixed (4), and 3 types = cirrus (5), shadow (6), adjacency (7); the parenthetical values correspond to the rows of the matrix shown in the "Units/Range" cell, the column represent the bands I1, I2, I3, M1, M2, M3, M4, M5, M7, M8, M10, and M11 in this order.	Unitless/ 0.0 ≤ cloud_wgts ≤ 1.0 cloud_wgts = (**I1,I2,I3,M1,M2,M3,M4, M5,M7,M8,M10,M11**) [0.5;**(1)** 0.5;**(2)** 0.5;**(3)** 0.5;**(4)** 0.6;**(5)** 0.3;**(6)** 0.8]**(7)**
cot_gy	float*32 x (nbands_i + nbands_m) x num_cloud_types	Cloud Optical Thickness "GREEN/YELLOW" quality threshold values	Unitless/ cot_gy = 0.2
cot_yr	float*32 x (nbands_i + nbands_m) x num_cloud_types	Cloud Optical Thickness "YELLOW/RED" quality threshold values	Unitless/ cot_yr = 0.5
qwgt_r	float*32 x (nbands_i + nbands_m)	Solar Zenith Angle boundaries "RED"	Unitless/ qwgt_r = 0.3
qwgt_y	float*32 x (nbands_i + nbands_m)	Solar Zenith Angle boundaries "YELLOW"	Unitless/ qwgt_y = 0.5
qwgt_g	float*32 x (nbands_i + nbands_m)	Solar Zenith Angle boundaries "GREEN"	Unitless/ qwgt_g = 0.7
frac_wgt_yr	float*32	Fractional Weight "YELLOW/RED" Threshold	Unitless/ ***Currently frac_wgt_yr = 0.4***
fract_wgt_gy	float*32	Fractional Weight "GREEN/YELLOW" Threshold	Unitless/ ***Currently frac_wgt_gy = 0.6***

Table 7. Atmospheric Intrinsic Reflectance LUT

Input	Data Type/Size	Description/Source	Units/Range
iband	int*32	Number of bands	Unitless 1 ≤ iband ≤ 9
itau	int*32	Number of aerosol optical thickness	Unitless 1 ≤ itau ≤ 15
ival	int*32	Number of scattering angles	unitless 1 ≤ ival ≤ 8000
rolut	float*32 x 9 x 15 x 8000	Atmospheric reflectance	Unitless > 0

Note: LUT file names of form reslut-k1-$band-$aerosolmodel.hdf or reslut-k2-$band-$aerosolmodel.hdf are constructed within the code. Array dimension limits have been hardcoded.

Table 8. Atmospheric Spherical Albedo LUT

Input	Data Type/Size	Description/Source	Units/Range
iband	int*32	Number of bands	Unitless 1 ≤ iband ≤ 9
itau	int*32	Number of optical thicknesses	Unitless 1 ≤ itau ≤ 15
sphalb	float*32 x 9x 15	Atmospheric spherical albedo	Unitless 0 ≤ sphalb ≤ 1

Note: LUT file names of form AEROLUT-$band-$aerosolmode are constructed within the code. Array dimension limits have been hardcoded.

Table 9. Total Atmospheric Transmission LUT

Input	Data Type/Size	Description/Source	Units/Range
iband	int*32	Number of bands	Unitless 1 ≤ iband ≤ 9
itau	int*32	Number of aerosol optical thicknesses	Unitless 1 ≤ itau ≤ 15
ival	int*32	Number of zenith angles	unitless 1 ≤ ival ≤ 21
trans	float*32 x 9 x 15 x 21	Atmospheric transmittance	Unitless 0 ≤ trans ≤ 1

Note: Lut file names of form TRANS-$band-aerosolmode are constructed within the code. Array limits are hardcoded.

Table 10. Angle LUT

Input	Data Type/Size	Description/Source	Units/Range
tsmax	float*32	Max scattering angle	degrees
tsmin	float*32	Min scattering angle	degrees
ttv	float*32	View zenith angle	degrees
tts	float*32	Solar zenith angle	degrees
nbfi	float*32	number of angles	unitless
nbfic	float*32	number of angles	unitless
indts	int*32	Scattering angle index	unitless

Note: Lut file name is ANGLE_NEW.hdf is hardcoded within the code.

Table 11. BRDF-coupling K1 LUT (Li-Sparse-Reciprocal BRDF kernel based)

Input	Data Type/Size	Description/Source	Units/Range
n_phi	int*8	Number of scattering angle bin values	Untiless/ ***(Currently set to 40)***
n_vza	int*8	Number of View Zenith Angle bin values	Untiless/ ***(Currently set to 21)***
n_sza	int*8	Number of Solar Zenith Angle bin values	Untiless/ ***(Currently set to 21)***
n_aot_bin	int*8	Number of AOT Bin Values	Untiless/ ***(Currently set to 15)***
AOT_VALUES	float*32 x n_aot_bin	AOT Bin Values	Untiless/ 0.0 ≤ AOT_VALUES ≤ 2.0

Input	Data Type/Size	Description/Source	Units/Range
robark1lut	float*32 x nbands_m x n_phi n_vza x n_sza x n_aot_bin	Theoretical Reflectance @ 550nm computed by convolving downward radiance with Li-Sparse Reciprocal BRDF kernel	Unitless 0.0 ≤ robark1lut ≤ 1.0

* LUT file names of form reslutrobar-k1-$bandname-$aerosolmodel.hdf are constructed within the code.

Table 12. BRDF-coupling K2 LUT (Ross-Thick BRDF kernel based)

Input	Data Type/Size	Description/Source	Units/Range
n_phi	int*8	Number of scattering angle bin values	Untiless/ ***(Currently set to 40)***
n_vza	int*8	Number of View Zenith Angle bin values	Untiless/ ***(Currently set to 21)***
n_sza	int*8	Number of Solar Zenith Angle bin values	Untiless/ ***(Currently set to 21)***
n_aot_bin	int*8	Number of AOT Bin Values	Untiless/ ***(Currently set to 15)***
AOT_VALUES	float*32 x n_aot_bin	AOT Bin Values	Unitless/ 0.0 ≤ AOT_VALUES ≤ 2.0
robark2lut	float*32 x nbands_m x n_phi n_vza x n_sza x n_aot_bin	Theoretical Reflectance @ 550nm computed by convolving downward radiance with Ross-Thick BRDF kernel	Unitless 0.0 ≤ robark1lut ≤ 1.0

* LUT file names of form reslutrobar-k2-$bandname-$aerosolmodel.hdf are constructed within the code.

Table 13. Atmospheric Correction Coefficients LUT

Input	Data Type/Size	Description/Source	Units/Range
oztransa	float*32 x nbands_m	Ozone Transmission Coefficients for all moderate resolution bands	Unitless/ oztransa = [-9.6905e-05,-3.4274e-03, -2.2472e-02,-9.5607e-02, -4.0222e-02,-1.4304e-05, 0.0,0.0,0.0]
wvtransa	float*32 x nbands_m	Water Vapor Transmission Coefficients for all moderate resolution bands	Unitless/ wvtransa = [-6.5405e-06,-3.1982e-06, -2.5304e-06,-4.4757e-04, -4.9331e-04,-3.1748e-03, -8.8228e-04,-8.8223e-04, -2.1476e-03]
wvtransb	float*32 x nbands_m	Water Vapor Transmission Coefficients for all moderate resolution bands	Unitless/ wvtransb = [-2.3931e-05,-1.5229e-05, -1.0287e-05, 2.1183e-04, -1.6048e-04,-3.8327e-03, 4.0951e-05, 4.0914e-05, -1.0312e-03]
wvtransc	float*32 x nbands_m	Water Vapor Transmission Coefficients for all moderate resolution bands	Unitless/ wvtransc = [1.3307e-06,6.5877e-07, 5.1815e-07,6.5553e-05, 7.3547e-05,5.3954e-04, 1.0347e-04,1.0346e-04, 3.4825e-04]

Input	Data Type/Size	Description/Source	Units/Range
ogtransa0	float*32 x nbands_m	Other Gases Transmission Coefficients for all moderate resolution bands	Unitless/ ogtransa0 = [-2.2785e-05, 1.4132e-05, -3.1599e-07,-2.9160e-06, -8.5001e-03,-4.6244e-05, -1.1720e-03,-1.9292e-02, -3.9956e-02]
ogtransa1	float*32 x nbands_m	Other Gases Transmission Coefficients for all moderate resolution bands	Unitless/ ogtransa1 = [-1.2939e-05, 6.2694e-06, 3.9837e-07,-1.1723e-06, -3.0282e-05,-4.9404e-06, -4.7744e-05,-2.5320e-05, -4.1868e-04]
ogtransb0	float*32 x nbands_m	Other Gases Transmission Coefficients for all moderate resolution bands	Unitless/ ogtransb0 = [3.3084e-05,-3.3714e-05, 2.0821e-06, 7.5342e-06, 6.3863e-03, 3.2619e-05, 2.1155e-04. 1.3167e-03, -1.1236e-02]
ogtransb1	float*32 x nbands_m	Other Gases Transmission Coefficients for all moderate resolution bands	Unitless/ ogtransb1 = [2.5561e-05,-1.5437e-05, -4.1673e-07, 3.0433e-06, 1.4216e-04, 5.8417e-06, 7.4862e-06,-3.5059e-05, 5.8990e-05]
ogtransc0	float*32 x nbands_m	Other Gases Transmission Coefficients for all moderate resolution bands	Unitless/ ogtransc0 = [5.8724e-06,-3.3437e-06, -4.8433e-07, 2.2723e-07, 2.2232e-03, 9.4750e-06, 2.3811e-04, 3.7652e-03, 6.3202e-03]
ogtransc1	float*32 x nbands_m	Other Gases Transmission Coefficients for all moderate resolution bands	Unitless/ ogtransc1 = [3.2050e-06,-1.3062e-06, -2.7715e-07, 8.6054e-08, 1.8262e-05, 5.1841e-07, 9.8521e-06, 1.5130e-05, 1.6264e-04]
tauray	float*32 x nbands_m	Atmospheric Intrinsic Reflectance Coefficients for all moderate resolution bands	Unitless/ tauray = [0.31233,0.22613,0.15334, 0.09243,0.04267,0.01551, 0.00364,0.00128,0.00034]

Table 14. Ross-Thick Volumetric Scattering LUT

Input	Data Type/Size	Description/Source	Units/Range
whitesky_RT	float*32	Ross-Thick Volumetric Scattering LUT	Unitless/ ***(Currently set to 0.18918039)***

Note: Code currently reads only the first value from the LUT table file

Table 15. Reciprocal Li-Sparse Geometric-Optical Scattering LUT

Input	Data Type/Size	Description/Source	Units/Range
whitesky_LSR	float*32	Reciprocal Li-Sparse Geometric-Optical Scattering LUT	Unitless/ ***(Currently set to -1.37719762)***

Note: Code currently reads only the first value from the LUT table file

Table 16. Snow Reflectance LUT

Input	Data Type/Size	Description/Source	Units/Range
nbands_m	int*32	Number of Moderate Resolution Bands	Unitless/ nbands_m = 9
band	char*8	String values for 9 Moderate Bands	Unitless/ band = `[M1,M2,M3,M4,M5, M7,M8,M10,M11]`
n_soot	int*32	Number of soot concentration bin values	Unitless/ n_soot = 4
soot_concentration	float*32 x n_soot	Soot Concentration Bin values	ppmw soot_concentration = `[0, 1, 10, 100]`
n_snow_grain	int*32	number of snow grain bin values	Unitless/ n_snow_grain = 4
snow_grain_size	float*32 x n_snow_grain	Snow Grain Size Bin Values	µm/ snow sizes = `[50, 100, 250, 500, 750, 1000]`
n_ami	int*32	Number of Aerosol Models	Unitless/ n_ami = 2
ami_bin	int*32 x n_ami	AMI Bin Values	Unitless/ ami_bin[1] = 1 (6s-Continental) ami_bin[2] = 2 (6s-Maritime)
n_aot	int*32	Number of AOT Bin Values	Unitless/ n_aot = 4
aot_bins	float*32 x n_aot	AOT Bin Values	Unitless/ 0.0 ≤ aot_bins ≤ 1.0 aot_bins = `[0., 0.2366, 0.5472, 1.0]`
n_wvot	int*32	Number of Precipitable Water Bin Values	Unitless/ nwvot = 3
wvot	float*32 x n_wvot	Precipitable Water Bin Values	gm/cm^2/ wvot = `[0.0, 0.4323, 2.0]`
n_oot	int*32	Number of Ozone Column Bin Values	Unitless/ n_oot = 3
oot	float*32 x n_oot	Ozone Column Bin Values	cm-atm/ oot = `[0.0, 0.1967, 0.5]`
n_sza	Float Scalar	Number of Solar Zenith Angle Bin Values	Unitless/ n_sza = 11
sza_bins	Float(n_sva)	Solar Zenith Angle Bin Values	Unitless/ Equally spaced in cos(sza) from sza = 85:0 degrees

Input	Data Type/Size	Description/Source	Units/Range
n_sva	Float Scalar	Number of View Zenith Angle Bin Values	Unitless/ n_sva = 7
sva_bins	Float (n_lsva)	View Zenith Angle Bin Values	Unitless/ Equally spaced in cos(sva) from sva = 70:0 degrees
n_relaz	int*32	Number of Relative Azimuth Angle Bin Values	Unitless/ n_relaz = 7
relaz_bins	float*32 x n_relaz	Relative Azimuth Bin Values	Degrees/ relaz_bins = [0, 30, 60, 90,120,150,180]
snow_toa_reflectance	float*32 x n_ami x n_bands_m x n_soot x n_snow_grain x n_aot x n_wvot x n_oot x n_sza x n_sva x n_relaz)	Snow TOA Reflectances from 6s RTM	Unitless/ snow_toa_reflectance > 0.0 *FILL_VALUE = -999.9*

Table 17. Snow Cover EDR Input File Specifications

Input	Object/Format	Original Source
VIIRS Earth View 375-meter & 750-meter SDR	HDF	VIIRS SDR Module
VIIRS Top-Of-Atmosphere Reflectance (Imagery resolution & Moderate Resolution)	HDF	VIIRS SDR Module
VIIRS BRDF Arch-Type IP	HDF	
VIIRS Local Angles IP	HDF	
VIIRS Surface Albedo IP	HDF	VIIRS Surface Albedo IP Module
VIIRS Aerosol Optical Thickness IP (AOT)	HDF	VIIRS AOT IP Module
VIIRS Aerosol Model Index IP (AMI)	HDF	VIIRS AMI IP Module
VIIRS Cloud Mask IP (VCM)	HDF	VIIRS Cloud Mask IP Module
VIIRS Cloud Optical Thickness IP (COT)	HDF	VIIRS Cloud Optical Thickness IP Module
NCEP Ozone Column	HDF	Ancillary Data Processing
NCEP Total Precipitable Water (TPW)	HDF	Ancillary Data Processing
NCEP Surface Pressure	HDF	Ancillary Data Processing
Snow Cover LUT	ASCII	Lookup Table Generation
Snow Cover Quality LUT	ASCII	Lookup Table Generation
Atmospheric Intrinsic Reflectance (BRDF-Coupling) LUT	HDF	Lookup Table Generation
Atmospheric Spherical Albedo LUT	HDF	Lookup Table Generation
Total Atmospheric Transmission LUT	HDF	Lookup Table Generation
Angle LUT	HDF	Lookup Table Generation
Atmospheric Correction Coefficients LUT	ASCII	Lookup Table Generation
Ross-Thick Volumetric Scattering LUT	ASCII	Lookup Table Generation
Reciprocal Li-Sparse Geometric-Optical Scattering LUT	ASCII	Lookup Table Generation
Snow Reflectance LUT	HDF	Lookup Table Generation

2.2.2 Outputs

2.2.2.1 Snow Cover EDR Output

The Snow Cover/Fraction EDR outputs are broken into two sections: 1) Outputs as a result of the spectral mixing algorithm and 2) Outputs as a result of aggregating the binary snow map output (Table 18. Binary Snow Map Output). The binary snow map is independent of the snow fraction algorithm, therefore the results will not change when switching between snow fraction modules.

Table 18. Binary Snow Map Output

Input	Data Type/Size	Description/Source	Units/Range
BinaryMap	int*8 x IBASCANS x IBATRACKS	Snow Binary Map	Unitless/ 0 = No Snow 1 = Snow *FILL_VALUE = -1*
BinaryMapQual	int*8 x IBASCANS x IBATRACKS	See Table 20	See Table 20
Latitude_Img	float*32 x IBASCANS x IBATRACKS	Latitude @ Imagery Resolution	Degrees/ -90 ≤ Latitude_Img ≤ 90 *FILL_VALUE = -999.9*
Longitude_Img	float*32 x IBASCANS x IBATRACKS	Longitude @ Imagery Resolution	Degrees/ -180 ≤ Longitude_Img ≤ 180 *FILL_VALUE = -999.9*

2.2.2.1.1 Snow Fraction (Multiple Endmember Spectral Mixture)

This output is produced using all 9 of the VIIRS Moderate Resolution bands and numerous LUT described in Table 5 - Note: Code currently reads only the first value from the LUT table file Table 16. IDPS should not convert code associated with spectral mixture method for operational use. NGST has commented out branches that invoke the spectral mixture method. IDPS should retain the delivered code with the spectral mixture method for possible future use. The following spectral mixture method snow fraction output parameters described in Table 19 will not produced as output but are documented here for possible future use.

Table 19. Snow Fraction EDR Output (Multiple Endmember Spectral Mixture Algorithm)

Input	Data Type/Size	Description/Source	Units/Range
SnowFraction	float*32 x MBASCANS x MBATRACKS	Snow Fraction reported at Moderate Resolution.	Unitless/ 0.0 ≤ SnowFraction ≤ 1.0 *FILL_VALUE = -1.0*
FractionWgt	float*32 x MBASCANS x MBATRACKS	Snow Fraction Weight	Unitless/ 0.0 ≤ FractionWgt ≤ 1.0
FractionQual	uint*8 x MBASCANS x MBATRACKS	**See** **Table 21**	**See** **Table 21**
SnowGrainSize	float*32 x MBASCANS x MBATRACKS	Snow Grain Size	μm <= 0 SnowGrainSize < 1000.0
SnowImpurity	float*32 x MBASCANS x MBATRACKS	Snow Soot Concentration	ppmw >0 [0
FractionFitError	float*32 x MBASCANS x MBATRACKS	Snow Fraction least squares fit error	Unitless >0.0
Latitude_Mod	float*32 x MBASCANS x MBATRACKS	Latitude @ Moderate Resolution	Degrees/ -90 ≤ Latitude_Mod ≤ 90 *FILL_VALUE = -999.9*

Input	Data Type/Size	Description/Source	Units/Range
Longitude_Mod	float*32 x IBASCANS x IBATRACKS	Longitude @ Moderate Resolution	Degrees/ -180 ≤ Longitude_Mod ≤ 180 *FILL_VALUE = -999.9*

Table 20. Binary Map Quality Mask Bit Description (375M Resolution)

BYTE	Bit	Flag Description Key	Result
0	0-1	Overall Pixel Quality	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED (POOR)
	2	Land/Water Background	0 = Land 1 = Water
	3	Forest	0 = No 1 = Yes
	4	Coastline	0 = No 1 = Yes
	5-6	Cloud Quality (Result of VCM cloud confidence and cloud quality test)	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED((POOR)
	7	Thin Cirrus (Solar)	0 = No 1 = Yes
1	0	Thin Cirrus (IR)	0 = No 1 = YES
	1-2	Cloud Phase	00 = Clear 01 = Water 10 = Ice 11 = Mixed
	3	Cloud Shadow	0 = No 1 = Yes
	4	Fire/Sun Glint	0 = No 1 = Yes
	5	Thermal Mask Quality	0 = GOOD 1 = POOR
	6	NDSI Quality	0 = GOOD 1 = POOR
	7	NDVI Quality	0 = GOOD 1 = POOR

Table 21. Snow Fraction Quality Mask Bit Description (750M Resolution)

BYTE	Bit	Flag Description Key	Result
0	0-1	Overall Pixel Quality	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED (POOR)
	2	Land/Water Background	0 = Land 1 = Water
	3	Forest	0 = No 1 = Yes
	4	Coastline	0 = No 1 = Yes
	5-6	Cloud Quality (Result of VCM cloud confidence and cloud quality test)	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED((POOR)
	7	Thin Cirrus (Solar)	0 = No 1 = Yes
1	0	Thin Cirrus (IR)	0 = No 1 = YES
	1-2	Cloud Phase	00 = Clear 01 = Water 10 = Ice 11 = Mixed
	3	Cloud Shadow	0 = No 1 = Yes
	4	Fire/Sun Glint	0 = No 1 = Yes
	5	Snow Reflectance Quality	0 = GOOD 1 = POOR
	6	Non-Snow Reflectance Quality	0 = GOOD 1 = POOR
	7	Pixel Weight Quality	0 = GOOD 1 = POOR

2.2.2.1.2 Snow Fraction (Binary Snow Map Aggregation)

This production of this output only requires the counting up of imagery snow pixels and determining a snow fraction for each moderate resolution pixel. The tuneable parameter fract_opt in the snow_cover.lut run configuration file should be set equal to 1 to select this option. Code associated with this option should be converted to operational code.

Table 22. Snow Fraction Output EDR (Binary Snow Map Aggregation)

Input	Data Type/Size	Description/Source	Units/Range
FractionFromBinaryMap	float*32 x MBASCANS x MBATRACKS	Snow Fraction reported at Moderate Resolution.	Unitless/ 0.0 ≤ FractionFromBinaryMap ≤ 1.0 *FILL_VALUE = -999.0*
NumAggPix	int*8 x MBASCANS x MBATRACKS	Number of Imagery Pixels Aggregated	Unitless/ 0≤ NumAggPix ≤ 4

2.2.3 I/O Timeliness Requirements

[*This section should list any timing requirements associated with any input and output data required for the EDR processing as well as for the overall processing chain/architecture that requires timely outputs from this EDR algorithm module. This information can be found in the ICD, and should include general discussions on data updated at regular intervals and data that are updated on infrequent intervals.*]

2.2.3.1 Requirements for Input

2.2.3.2 Requirements for Applicable Auxiliary/Ancillary and/or optional Input Data

None.

2.2.3.3 Requirements for Output

[Provide descriptions of output timeliness requirements, if any. Such requirements may include considerations for the downstream processing for other EDRs that may require output from this module as required or supporting input data.]

2.3 Overall Methodology

The snow cover EDR algorithm computes two datasets: a snow cover binary map (375M) and snow fraction (750M). Although the algorithms that produce the binary map and snow fraction are encapsulated in one large Fortran 90 routine (SN_main.f), the snow fraction output does not depend on the binary map and vice-versa. The binary map algorithm only requires a few LUT table parameters, various VCM bits, and the quality of the imagery reflectance bands I1, I2, and I3; the binary map logic and computation are described in section 2.3.1. The spectral mixture based snow fraction calculation is described in sections 2.3.1.2 - 2.3.2.4; another approach to compute the snow fraction is detailed in section 2.3.3. The spectral mixture based approach has been commented out and IDPS should not convert the spectral mixture related branches of the code to operatonal code. The overall flow of the Snow Cover EDR algorithm is shown in Figure 2.

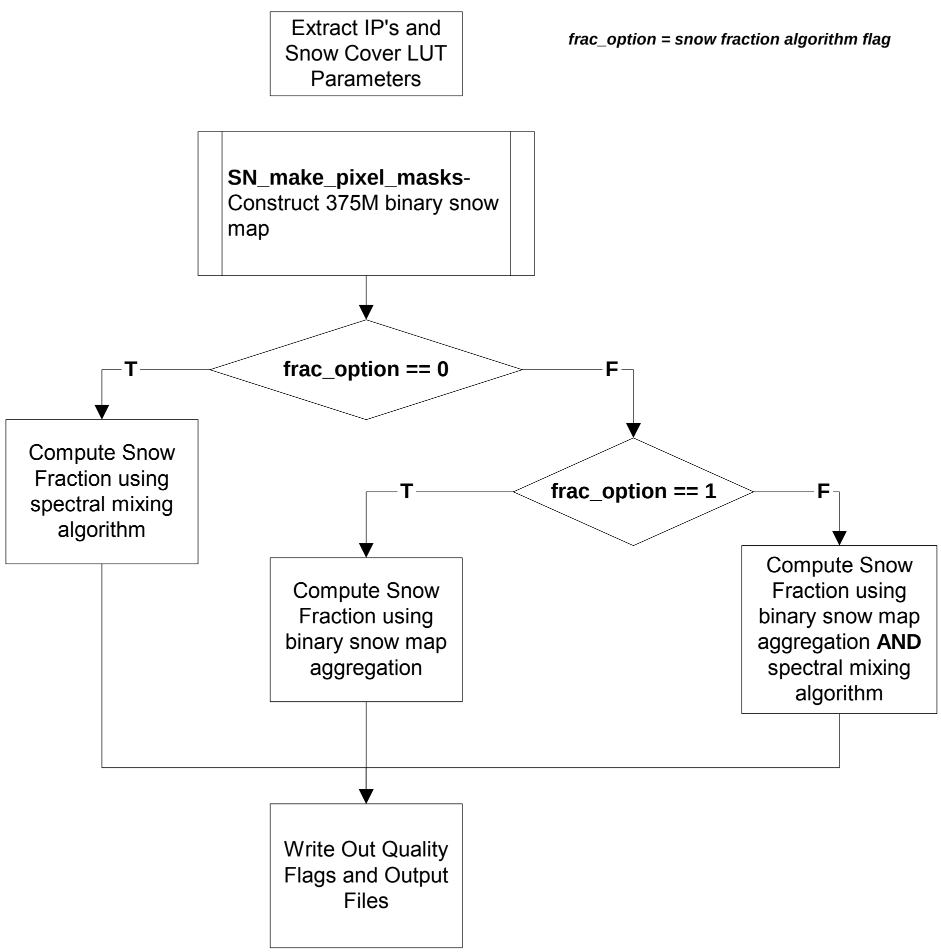

Figure 2. Overall Logic Flow of the Snow Cover EDR Module

2.3.1 Binary Snow Map

2.3.1.1 Binary Map (`SN_binary_map()`)

The binary map calculation requires inputs from the VCM bits and LUT's described in Table 3 and Table 5 respectively. Furthermore, the code requires the quality of each imagery reflectance band. The band quality and VCM bits are stored in a variable "mask_i" in a total of 3 bytes; Table 24 gives the bit description of "mask_i." The bits in Table 24 are constructed in a function called "**`SN_make_pixel_masks()`**"; this function is described in 2.3.1.2.

Using the bit description and the LUT parameters a snow binary map is constructed using the logic detailed in Figure 3; Table 23 describes variables used in Figure 3.

Table 23. Relevant Parameters in `SN_binary_map()`

Parameter	Description
toa_refl_i	TOA Reflectance for Imagery Bands I1, I2, and I3
P	Current Pixel
binary_map	Binary Snow Map
ndvi	Normalized Difference Vegetation Index (NDVI)
ndsi	Normalized Difference Snow Index (NDSI)
ndvi_quality	NDVI Quality (0 = GOOD/GREEN, 1 = POOR/RED)
ndsi_quality	NDSI Quality (0 = GOOD/GREEN, 1 = POOR/RED)

The rest of the parameters used in the flow chart are described in Table 5. The algorithm, in its present form, outputs the binary map within the function **`SN_binary_map()`**. The quality of each binary map pixel is determined and written out in the function **`SN_write_edr()`**,detailed in section 2.3.2.3. The binary map quality segment of **`SN_write_edr()`** can be moved to **`SN_binary_map()`** without affecting the output file contents.

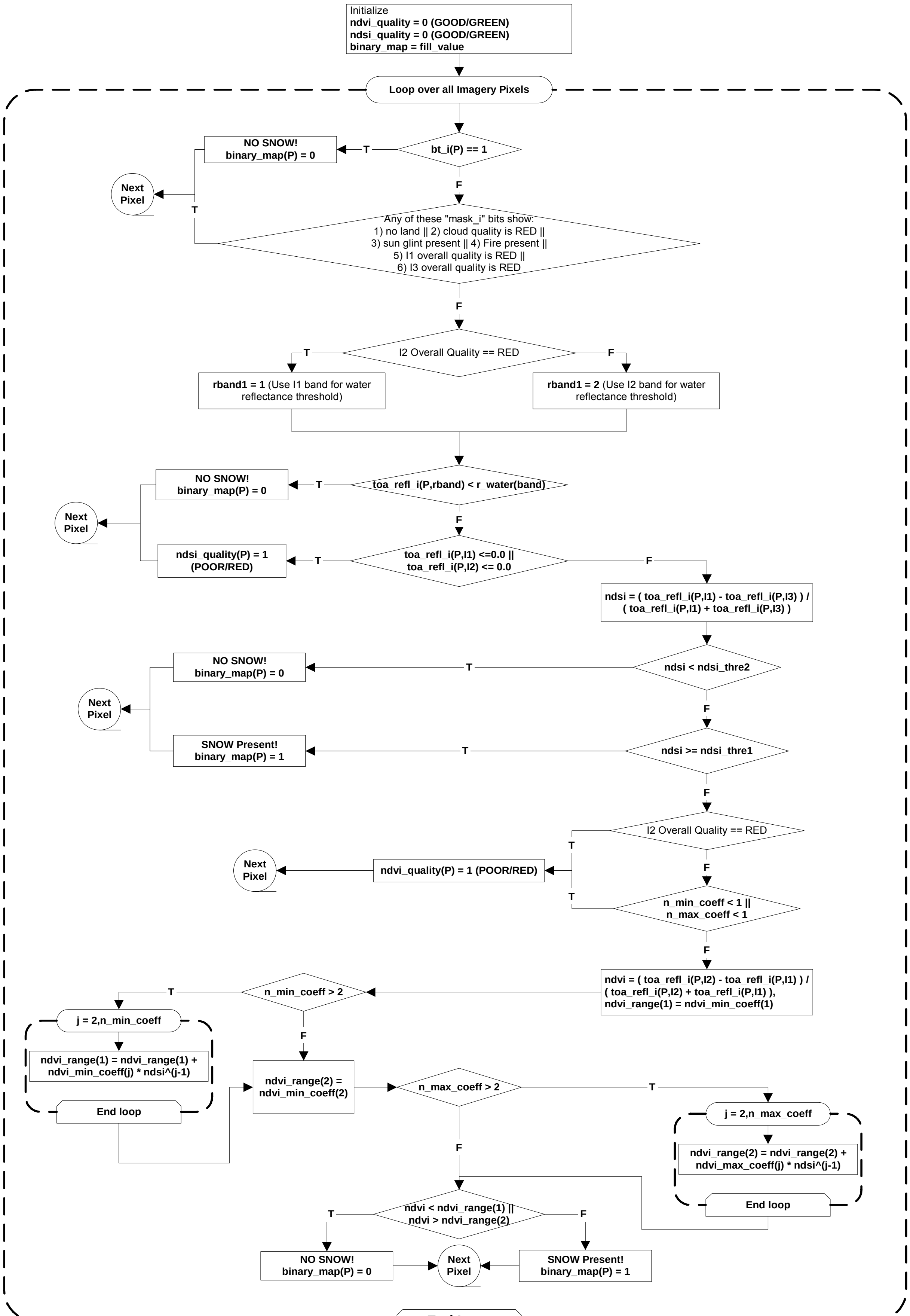

Figure 3. Logic Flow of `SN_binary_map()`

2.3.1.2 Pixel Mask Construction (`SN_make_pixel_masks()`)

This function takes the VCM and constructs quality bits, stored in the variable “mask_i” and “mask_m”, for the purpose of determining the binary snow map quality and snow fraction quality respectively.

Table 24. Description of VCM and Band Quality Bits (mask_i)

BYTE	Bit	Flag Description Key	Result
0	0	Land/Water Background	0 = Land 1 = Water
	1	Forest	0 = No 1 = Yes
	2	Coastline	0 = No 1 = Yes
	3-4	Cloud Quality (Result of VCM cloud confidence and cloud quality test)	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED((POOR)
	5	Thin Cirrus (Solar)	0 = No 1 = Yes
	6	Thin Cirrus (IR)	0 = No 1 = Yes
	7	Cloud Shadow	0 = No 1 = Yes
1	0-1	Cloud Phase	00 = Clear 01 = Water 10 = Ice 11 = Mixed
	2	Fire	0 = No 1 = Yes
	3	Sun Glint	0 = No 1 = Yes
	4	Imagery Band Quality (I1)	0 = GREEN (GOOD) 1 = RED (POOR)
	5	Imagery Band Quality (I2)	0 = GREEN (GOOD) 1 = RED (POOR)
	6	Imagery Band Quality (I3)	0 = GREEN (GOOD) 1 = RED (POOR)
	7	Spare	Spare
2	0-1	I1 Overall Quality	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED (POOR)
	2-3	I2 Overall Quality	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED (POOR)
	4-5	I3 Overall Quality	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED (POOR)

Table 25. Description of VCM and Band Quality Bits (mask_m)

BYTE	Bit	Flag Description Key	Result
0	0	Land/Water Background	0 = Land 1 = Water
	1	Forest	0 = No 1 = Yes
	2	Coastline	0 = No 1 = Yes
	3-4	Cloud Quality (Result of VCM cloud confidence and cloud quality test)	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED((POOR)
	5	Thin Cirrus (Solar)	0 = No 1 = Yes
	6	Thin Cirrus (IR)	0 = No 1 = Yes
	7	Cloud Shadow	0 = No 1 = Yes
1	0-1	Cloud Phase	00 = Clear 01 = Water 10 = Ice 11 = Mixed
	2	Fire	0 = No 1 = Yes
	3	Sun Glint	0 = No 1 = Yes
	4	Moderate Band Quality (M1)	0 = GREEN (GOOD) 1 = RED (POOR)
	5	Moderate Band Quality (M2)	0 = GREEN (GOOD) 1 = RED (POOR)
	6	Moderate Band Quality (M3)	0 = GREEN (GOOD) 1 = RED (POOR)
	7	Moderate Band Quality (M4)	0 = GREEN (GOOD) 1 = RED (POOR)
2	0	Moderate Band Quality (M5)	0 = GREEN (GOOD) 1 = RED (POOR)
	1	Moderate Band Quality (M7)	0 = GREEN (GOOD) 1 = RED (POOR)
	2	Moderate Band Quality (M8)	0 = GREEN (GOOD) 1 = RED (POOR)
	3	Moderate Band Quality (M10)	0 = GREEN (GOOD) 1 = RED (POOR)
	4	Moderate Band Quality (M11)	0 = GREEN (GOOD) 1 = RED (POOR)
	5-6	Overall M1 Quality	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED (POOR)
	7	Overall M2 Quality	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED (POOR)
3	0	Overall M2 Quality	
	1-2	Overall M3 Quality	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED (POOR)
	3-4	Overall M4 Quality	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED (POOR)
	5-6	Overall M5 Quality	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED (POOR)

BYTE	Bit	Flag Description Key	Result
	7	Overall M7 Quality	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED (POOR)
4	0	Overall M7 Quality	
	1-2	Overall M8 Quality	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED (POOR)
	3-4	Overall M10 Quality	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED (POOR)
	5-6	Overall M11 Quality	00 = GREEN (GOOD) 01 = GREEN/YELLOW 10 = YELLOW/RED 11 = RED (POOR)
	7	Overall Quality	0 = GREEN (GOOD) 1 = RED (POOR)

2.3.2 Snow Fraction (Multiple End-member Spectral Mixture Algorithm)

This algorithm uses all nine moderate resolution bands, as well as other inputs described in 2.2.1, to determine the snow fraction. The code is currently not ready for operational use but can be executed by setting a flag to run this branch of the snow fraction algorithm. NGST has commented out the branches related to the spectral mixture method. IDPS should not implement code associated with branches for fract_opt = 0 or fract_opt = 2 which have been commented out for operational use. IDPS should retain the commented spectral mixture method code for possible future use.

2.3.2.1 Snow Fraction (SN_snow_fraction())

The detailed logic of the function **`SN_snow_fraction()`** is not available at this time.
This subroutine should not be converted to operational code as it relates to the spectral mixture method.

2.3.2.2 Selection Snow Fraction Per Snow Type (`SN_select()`)

The detailed logic of the function **`SN_select()`** is not available at this time.
This subroutine should not be converted to operational code as it relates to the spectral mixture method.

2.3.2.3 Non-Snow Reflectance (`SN_nonsnow_reflectance()`)

The details of the non-snow reflectance computation in **`SN_nonsnow_reflectance()`** is not available at this time. This subroutine should not be converted to operational code as it relates to the spectral mixture method.

2.3.2.4 Snow Reflectance (SN_snow_reflectance())

The details of the snow reflectance computation in **`SN_snow_reflectance()`** is not available at this time. This subroutine should not be converted to operational code as it relates to the spectral mixture method.

2.3.3 Snow Fraction-(Binary Snow Map Aggregation)

This algorithm computes fractional snow cover by performing a 2x2 aggregation of the imagery resolution snow binary map. The snow fraction measurement uncertainty using this technique will be 25%. Aggregation is performed by counting the number of snow filled pixels within each 2x2 pixel aggregation window of the 375m resolution snow map, and dividing by the total number of pixels. Snow fraction (@ 750m resolution) is determined within to 25% measurement uncertainty. The “frac_option” flag (last entry of the snow_cover.lut configuration file) set to a value of 1 selects this branch of the snow cover algorithm. IDPS should convert code associated with the binary Snow Map Aggregation method of computing snow fraction.

2.3.4 Output EDR (`SN_write_edr()`)

This function serves to write out the quality flag parameters for both the snow binary map and snow fraction EDR’s. Note that this function does not write out the actual snow map and snow fraction parameters, these are produced in the functions described in sections 2.3.1 and 2.3.2 respectively. And since the snow binary map and snow fraction algorithms are independent of each other, i.e. no intermediate parameters produced by either code will act as an input parameter to the other software module, one can take the snow binary map portion of this function and move it to **`SN_binary_map()`**. Likewise, the snow fraction segment of this module can be transferred to the **`SN_snow_fraction()`** module.

Having said this, the additional parameters produced, within **`SN_write_edr()`**, for the snow binary map and snow fraction are enumerated as follows:

1. Snow Binary Map Quality Flags – See Table 20 for bit description. Almost all tests determining the binary map quality depends on the bits of “mask_i” (see Table 24). However, the NDVI, NDSI, and thermal mask (imagery band brightness temperature) qualities are required as well.
2. Snow Fraction Quality Flags - See

3.
4. Table 21 for bit description. Essentially all tests are dependent on the bits described in Table 25. However, the LUT parameter “frac_wgt_gy” and weighting factors “fraction_wgt” (determined by the quality of IP’s and LUT parameters in Table 6) are required as well.

Note: These parameters correspond to the spectral mixing snow fraction algorithm only. The rest of the output snow fraction parameters are written out in the function **`SN_select()`** (See section 2.3.2.2). As for the binary map quality determination, Figure 4 details this. Figure 5 describes the flow of the snow fraction quality algorithm.

This module, currently also outputs a snow fraction based on aggregating the binary snow map. The snow fraction output parameters for this branch of the algorithm are described in Table 22.

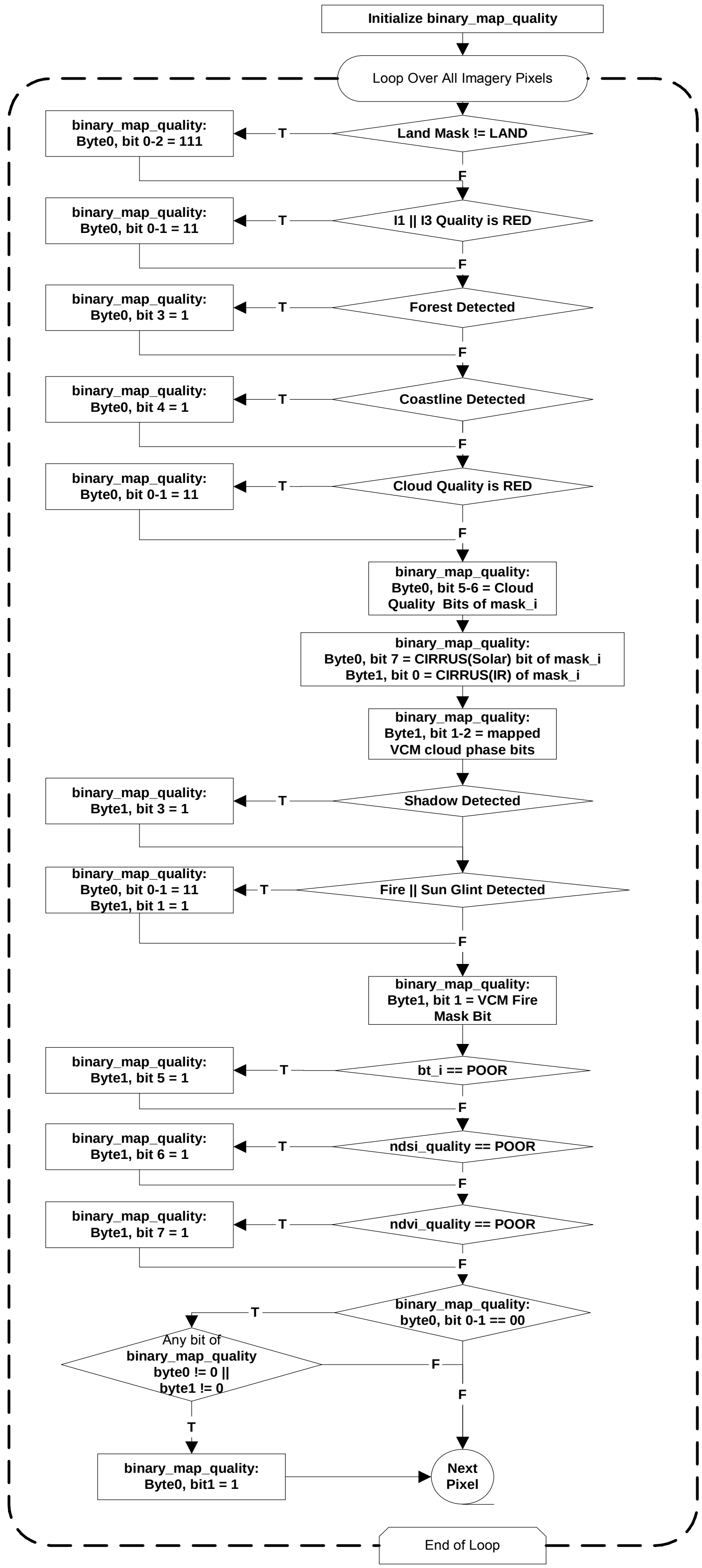

Figure 4. Logic Flow of the Binary Map Quality Portion of `SN_write_edr()`

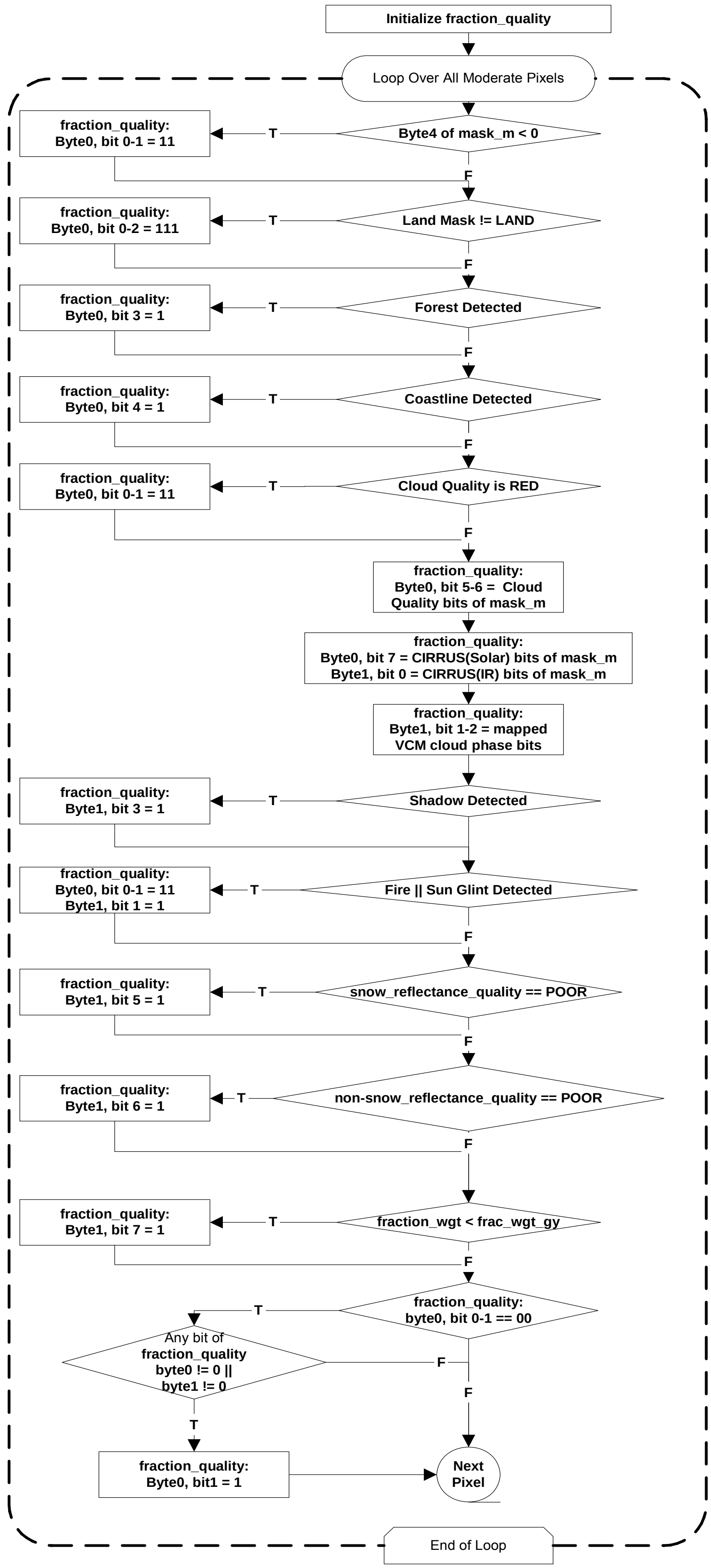

Figure 5. Logic Flow of the Snow Fraction Quality Portion of `SN_write_edr()`

3. Computational Precision Requirements

Single precision floating point computations are required by the snow cover algorithm.

4. Practical Implementation Considerations

Currently, the program is directed to stop upon encountering any out of range values associated with input IP and ancillary data files. Appropriate recovery procedures such as setting a fill value and cycling past invalid IP data should be implemented where appropriate.

4.1 Numerical Computation Considerations

[Provide any information/considerations that may be particular/relevant to the implementation of the algorithm. Such details may include discussions of coefficients or lookup table entries used by the implemented algorithm that may have to be generated offline prior to the actual execution of the algorithm. The algorithm coefficients may also require periodic updates. Discussions on such update frequency should be noted here as well.]

The spectral mixture method for computation of snow fraction is computationally efficient and is not expected to have latency concerns. However, it requires several lookup tables and input IP data files. The lookup tables are generated by offline processes but require no periodic updating. The look up tables and IP data inputs used specifically by the spectral mixture method for computation of snow fractions are summarized in Table 26.

Table 26. Spectral Mixture Method Data Input Requirements

Input	Object/Format	Original Source
VIIRS 750m SDR (M1,M2, M3, M4, M5, M7, M8, M10, M11)	HDF	VIIRS SDR Module
VIIRS BRDF Arch-Type IP	HDF	VIIRS Surface Albedo IP Module
VIIRS Local Angles IP	HDF	**TBD**
VIIRS Surface Albedo IP	HDF	VIIRS Surface Albedo IP Module
VIIRS Aerosol Optical Thickness IP (AOT)	HDF	VIIRS AOT IP Module
VIIRS Aerosol Model Index IP (AMI)	HDF	VIIRS AMI IP Module
VIIRS Cloud Mask IP (VCM)	HDF	VIIRS Cloud Mask IP Module
VIIRS Cloud Optical Thickness IP (COT)	HDF	VIIRS Cloud Optical Thickness IP Module
NCEP Ozone Column	HDF	Ancillary Data Processing
NCEP Total Precipitable Water (TPW)	HDF	Ancillary Data Processing
NCEP Surface Pressure	HDF	Ancillary Data Processing
Atmospheric Intrinsic Reflectance LUTs	HDF	Lookup Table Generation
BRDF Coupling LUTs	HDF	Lookup Table Generation
Atmospheric Spherical Albedo LUTs	HDF	Lookup Table Generation
Total Atmospheric Transmission LUTs	HDF	Lookup Table Generation
Angle LUT	HDF	Lookup Table Generation
Atmospheric Correction Coefficients LUT	ASCII	Lookup Table Generation
Ross-Thick Volumetric Scattering LUT	ASCII	Lookup Table Generation
Reciprocal Li-Sparse Geometric-Optical Scattering LUT	ASCII	Lookup Table Generation
Modeled Snow TOA reflectance	HDF	Lookup Table Generation

4.2 Exception Handling

[Provide any notes on the exception handling, as applicable.]

4.3 Software Environment Considerations

[Provide specific software environment considerations important to the implementation and for comparison purposes with the software tests.]

4.4 Quality Assessment and Diagnostics

[Include any relevant discussions on the quality assessments and diagnostics that may provide critical information for the proper algorithm implementation.]

4.5 Tunable Parameters

This section describes all the snow cover EDR tunable parameters and thresholds. Tunable parameters are defined in the snow_cover.lut run configuration file and the snow_quality.lut quality weights and threshold specification files (tables 5 and 6 respectively). Tuneable parameter from both tables are presented below

Table 27. Tunable Parameters

Input	Data Type/Size	Description/Source	Units/Range
nbands_m	int*32	Number of moderate resolution bands	Unitless/ nbands > 0 ***(Currently set to 9)***
band_m	int*32 x nbands_m	Band Numbers (nbands_m in size)	Unitless/ `[1,2,3,4,5,7,8,10,11]`
num_r_water	int*32	Number of water reflectance thresholds (For I1 and I2)	Unitless/ num_r_water > 0 ***(Currently set to 2)***
r_water	float*32 x num_r_water	Water Reflectance Thresholds (For I1 and I2)	Unitless/ 0.0 ≤ r_water ≤ 1.0 r_water = `[0.10, 0.11]`
ndsi_thre1	float*32	First NDSI Threshold	Unitless/ ***(Currently set to 0.4)***
ndsi_thre2	float*32	Second NDSI Threshold	Unitless ***(Currently set to 0.1)***
n_max_coeff	int*32	Number of NDVI Maximum Coefficients	Unitless n_max_coeff > 0 ***(Currently set to 4)***
ndvi_max_coeff	float*32 x n_max_coeff	NDVI Maximum Coefficients	Unitless ndvi_max_coeff = `[-0.28,6.4,-12.0,10.0]`
n_min_coeff	int*32	Number of NDVI Minimum Coefficients	Unitless n_min_coeff > 0 ***(Currently set to 2)***
ndvi_min_coeff	float*32 x n_min_coeff	NDVI Minimum Coefficients	Unitless/ ndvi_min_coeff = `[0.32,-0.70]`
btmax	float*32	Brightness Temperature Threshold	Kelvin/ ***(Currently set to 283.0)***
ntypes	int*32	Number of Snow Types (6 grain size * 4 impurities = 24 types)	Unitless/ 1 ≤ ntypes ≤ 24 ***(Currently set to 24)***
frac_option	int*32	Flag which determines which snow fraction algorithm to run.	Untiless/ 0 = Spectral Mixing Algorithm **1 = Binary Snow Map Aggregation** 2 = Both **(Currently set to 1)**

Input	Data Type/Size	Description/Source	Units/Range
nbands_i	int*32	Number of Imagery Resolution Bands	Unitless/ nbands_i > 0 ***(Currently set to 3)***

band_wgt	float*32 x nbands_m	Default Moderate Resolution Band Weights	Unitless/
			0.0 ≤ band_wgt ≤ 1.0
			band_wgt =
			`[1.0,1.0,1.0,`
			`1.0,1.0,1.0,`
			`1.0,1.0,1.0]`

num_aot_bins	int*32	Number of AOT bins, which correspond to the number of AOT values used for thresholding (aot_bin, this table)	Unitless/ num_aot_bins > 0 ***(Currently set to 4)***
aot_bins	float*32 x num_aot_bins	AOT Bin Boundary Values	Unitless/ 0.0 ≤ aot_bin ≤ 1.0 aot_bin = `[0.0,0.15,0.5,1.0]`
num_thresh	int*32	Number of Solar Zenith Angle Thresholds	Unitless num_thresh > 0 ***(Currently set to 2)***
q_aot_sza	float*32 x (nbands_i + nbands_m) x num_aot_bins x num_thresh	Solar Zenith Angle values that correspond to the Solar Zenith Angle quality regimes (G/Y = "Green/Yellow", Y/R = "Yellow/Red", this corresponds to the "2" in the "Data Types/Size" column) and to the "aot_bin" values (**(1) -> 0.0,(2) -> 0.15,(3) -> 0.5,(4) -> 1.0**). The order for each num_aot_bin x num_thresh matrix of angles is: I1, I2, I3, M1, M2, M3, M4, M5, M7, M8, M10, and M11	Degrees/ -90 ≤ sza_bin ≤ 90 (**I1,I2,I3,M1,M2,M3,M4,** **M5,M7,M8,M10,M11**) **(G/Y) (Y/R)** `[75.0, 85.0;`**(1)** `70.0, 85.0;`**(2)** `65.0, 80.0;`**(3)** `60.0, 75.0]`**(4)** (Order of bands follows the scheme showed above)
cot_switch	int*32	Switch to flag the availability of the Cloud Optical Thickness IP	Unitless/ 0 = COT Not Available (Use VCM) 1 = COT Available (Use COT) ***(Currently set to 0)***
num_cloud_types	int*32	Number of Cloud Types	Unitless/ num_cloud_types > 0 ***(Currently set to 7)***
cloud_wgts	float*32 x num_cloud_types x nbands	Cloud weights corresponding to the 3 imagery bands + 9 moderate bands and the 7 cloud properties - 4 phases = Default (1), Water (2), Ice (3), Mixed (4), and 3 types = cirrus (5), shadow (6), adjacency (7); the parenthetical values correspond to the rows of the matrix shown in the "Units/Range" cell, the column represent the bands I1, I2, I3, M1, M2, M3, M4, M5, M7, M8, M10, and M11 in this order.	Unitless/ 0.0 ≤ cloud_wgts ≤ 1.0 cloud_wgts = (**I1,I2,I3,M1,M2,M3,M4,** **M5,M7,M8,M10,M11**) `[0.5;`**(1)** `0.5;`**(2)** `0.5;`**(3)** `0.5;`**(4)** `0.6;`**(5)** `0.3;`**(6)** `0.8]`**(7)**

Input	Data Type/Size	Description/Source	Units/Range
cot_gy	float*32 x (nbands_i + nbands_m) x num_cloud_types	Cloud Optical Thickness “GREEN/YELLOW” quality threshold values	Unitless/ cot_gy = 0.2
cot_yr	float*32 x (nbands_i + nbands_m) x num_cloud_types	Cloud Optical Thickness “YELLOW/RED” quality threshold values	Unitless/ cot_yr = 0.5
qwgt_r	float*32 x (nbands_i + nbands_m)	Solar Zenith Angle boundaries “RED”	Unitless/ qwgt_r = 0.3
qwgt_y	float*32 x (nbands_i + nbands_m)	Solar Zenith Angle boundaries “YELLOW”	Unitless/ qwgt_y = 0.5
qwgt_g	float*32 x (nbands_i + nbands_m)	Solar Zenith Angle boundaries “GREEN”	Unitless/ qwgt_g = 0.7
frac_wgt_yr	float*32	Fractional Weight “YELLOW/RED” Threshold	Unitless/ ***Currently frac_wgt_yr = 0.4***
fract_wgt_gy	float*32	Fractional Weight “GREEN/YELLOW” Threshold	Unitless/ ***Currently frac_wgt_gy = 0.6***

4.6 Program Parameters for Continuous monitoring

These parameters should be monitored throughout the course of the NPP and NPOESS program.

Table 28. Parameters for Continuous Monitoring

Parameter	Data Type/Size	Description/Source	Units/Range
Reflectance_Mod	float*32 x MBASCANS x MBATRACKS x nbands_m	TOA Reflectances for all Moderate Resolution Bands	Unitless/ 0.0 ≥ Reflectance_Mod *FILL_VALUE = 65535* (Integer Scaled)
Reflectance_Img	float*32 x IBASCANS x IBATRACKS	TOA Reflectances for all Imagery Resolution Bands	Unitless/ 0.0 ≥ Reflectance_Img *FILL_VALUE = 65535* (Integer Scaled)
BrightTemp_M15	float*32 x MBASCANS x MBATRACKS	Band M15 Brightness Temperatures	Kelvin/ 0.0 ≥ bt15_m *FILL_VALUE = 65535* (Integer Scaled)
BrightTemp_M16	float*32 x MBASCANS x MBATRACKS	Band M16 Brightness Temperatures	Kelvin/ 0.0 ≥ bt16_m *FILL_VALUE = 65535* (Integer Scaled)
BrightTemp_I5	float*32 x IBASCANS x IBATRACKS	Band I5 Brightness Temperatures	Kelvin/ 0.0 ≥ bt5_i *FILL_VALUE = 65535* (Integer Scaled)

5. Operational Code Verification Test Datasets

[Describe the usage of the "Operational Code Verification Test Datasets". This pertains to the datasets used to verify the operational quality performance.]

5.1 Test Procedure and Results

No additional tests beyond the unit tests are provided. Details regarding the unit tests are describted in the VIIRS Snow Cover EDR Science Grade Software Unit Test Document (No. NP-EMD.2005.510.0047)

6. Assumptions and Limitations

6.1 Assumptions

None identified at this time.

6.2 Limitations

- Retrievals will not be performed under nighttime conditions. This is defined as instances where the solar zenith angle exceeds 85 degrees.
- Retrievals will not be performed under confident cloudy conditions.
- Retrievals will not be performed over ocean or water surfaces.
- Retrievals of the binary snow map and snow fraction will be questionable under conditions of extreme aerosol loading, such as events associated with volcanic eruptions or biomass burning, are not guaranteed to meet the performance specification for these circumstances

7. Glossary/Acronym List

7.1 Glossary

Table 5. Glossary

TERM	DESCRIPTION
Algorithm	A formula or set of steps for solving a particular problem. Algorithms can be expressed in any language, from natural languages like English to mathematical expressions to programming languages like FORTRAN. On NPOESS, an algorithm consists of: 1. A theoretical description (i.e., science/mathematical basis) 2. A computer implementation description (i.e., method of solution) 3. A computer implementation (i.e., code)
Algorithm Configuration Control Board (ACCB)	Interdisciplinary team of scientific and engineering personnel responsible for the approval and disposition of algorithm acceptance, verification, development and testing transitions. Chaired by the Algorithm Implementation Process Lead, members include representatives from IWPTB, Systems Engineering & Integration IPT, System Test IPT, and IDPS IPT
Algorithm Verification	Science-grade software delivered by an algorithm provider is verified for compliance with data quality and timeliness requirements by Algorithm Team science personnel. This activity is nominally performed at the IWPTB facility. Delivered code is executed on compatible IWPTB computing platforms. Minor hosting modifications may be made to allow code execution. Optionally, verification may be performed at the Algorithm Provider's facility if warranted due to technical, schedule or cost considerations
Ancillary Data	Any data which is not produced by the NPOESS System, but which is acquired from external providers and used by the NPOESS system in the production of NPOESS data products.
Auxiliary Data	Auxiliary Data is defined as data, other than data included in the sensor application packets, which is produced internally by the NPOESS system, and used to produce the NPOESS deliverable data products.
EDR Algorithm	Scientific description and corresponding software and test data necessary to produce one or more environmental data records. The scientific computational basis for the production of each data record is described in an ATBD. At a minimum, implemented software is science-grade and includes test data demonstrating data quality compliance
Environmental Data Record (EDR)	*[IORD Definition]* Data record produced when an algorithm is used to convert Raw Data Records (RDRs) to geophysical parameters (including ancillary parameters, e.g., cloud clear radiation, etc.). *[Supplementary Definition]* An Environmental Data Record (EDR) represents the state of the environment, and the related information needed to access and understand the record. Specifically, it is a set of related data items that describe one or more related estimated environmental parameters over a limited time-space range. The parameters are located by time and Earth coordinates. EDRs may have been resampled if they are created from multiple data sources with different sampling patterns. An EDR is created from one or more NPOESS SDRs or EDRs, plus ancillary environmental data provided by others. EDR metadata contains references to its processing history, spatial and temporal coverage, and quality.
Operational Code	Verified science-grade software, delivered by an algorithm provider and verified by IWPTB, is developed into operational-grade code by the IDPS IPT.
Operational-Grade Software	Code that produces data records compliant with the System Specification requirements for data quality and IDPS timeliness and operational infrastructure. The software is modular relative to the IDPS infrastructure and compliant with IDPS application programming interfaces (APIs) as specified for TDR/SDR or EDR code

TERM	DESCRIPTION
Raw Data Record (RDR)	*[IORD Definition]* Full resolution digital sensor data, time referenced and earth located, with absolute radiometric and geometric calibration coefficients appended, but not applied, to the data. Aggregates (sums or weighted averages) of detector samples are considered to be full resolution data if the aggregation is normally performed to meet resolution and other requirements. Sensor data shall be unprocessed with the following exceptions: time delay and integration (TDI), detector array non-uniformity correction (i.e., offset and responsivity equalization), and data compression are allowed. Lossy data compression is allowed only if the total measurement error is dominated by error sources other than the data compression algorithm. All calibration data will be retained and communicated to the ground without lossy compression. *[Supplementary Definition]* A Raw Data Record (RDR) is a logical grouping of raw data output by a sensor, and related information needed to process the record into an SDR or TDR. Specifically, it is a set of unmodified raw data (mission and housekeeping) produced by a sensor suite, one sensor, or a reasonable subset of a sensor (e.g., channel or channel group), over a specified, limited time range. Along with the sensor data, the RDR includes auxiliary data from other portions of NPOESS (space or ground) needed to recreate the sensor measurement, to correct the measurement for known distortions, and to locate the measurement in time and space, through subsequent processing. Metadata is associated with the sensor and auxiliary data to permit its effective use.
Retrieval Algorithm	A science-based algorithm used to 'retrieve' a set of environmental/geophysical parameters (EDR) from calibrated and geolocated sensor data (SDR). Synonym for EDR processing.
Science Algorithm	The theoretical description and a corresponding software implementation needed to produce an NPP/NPOESS data product (TDR, SDR or EDR). The former is described in an ATBD. The latter is typically developed for a research setting and characterized as "science-grade".
Science Algorithm Provider	Organization responsible for development and/or delivery of TDR/SDR or EDR algorithms associated with a given sensor
Science-Grade Software	Code that produces data records in accordance with the science algorithm data quality requirements. This code, typically, has no software requirements for implementation language, targeted operating system, modularity, input and output data format or any other design discipline or assumed infrastructure
SDR/TDR Algorithm	Scientific description and corresponding software and test data necessary to produce a Temperature Data Record and/or Sensor Data Record given a sensor's Raw Data Record. The scientific computational basis for the production of each data record is described in an Algorithm Theoretical Basis Document (ATBD). At a minimum, implemented software is science-grade and includes test data demonstrating data quality compliance
Sensor Data Record (SDR)	*[IORD Definition]* Data record produced when an algorithm is used to convert Raw Data Records (RDRs) to calibrated brightness temperatures with associated ephemeris data. The existence of the SDRs provides reversible data tracking back from the EDRs to the Raw data. *[Supplementary Definition]* A Sensor Data Record (SDR) is the recreated input to a sensor, and the related information needed to access and understand the record. Specifically, it is a set of incident flux estimates made by a sensor, over a limited time interval, with annotations that permit its effective use. The environmental flux estimates at the sensor aperture are corrected for sensor effects. The estimates are reported in physically meaningful units, usually in terms of an angular or spatial and temporal distribution at the sensor location, as a function of spectrum, polarization, or delay, and always at full resolution. When meaningful, the flux is also associated with the point on the Earth geoid from which it apparently originated. Also, when meaningful, the sensor flux is converted to an equivalent top-of-atmosphere (TOA) brightness. The associated metadata includes a record of the processing and sources from which the SDR was created, and other information needed to understand the data.

TERM	DESCRIPTION
Temperature Data Record (TDR)	*[IORD Definition]* Temperature Data Records (TDRs) are geolocated, antenna temperatures with all relevant calibration data counts and ephemeris data to revert from T-sub-a into counts. *[Supplementary Definition]* A Temperature Data Record (TDR) is the brightness temperature value measured by a microwave sensor, and the related information needed to access and understand the record. Specifically, it is a set of the corrected radiometric measurements made by an imaging microwave sensor, over a limited time range, with annotation that permits its effective use. A TDR is a partially-processed variant of an SDR. Instead of reporting the estimated microwave flux from a specified direction, it reports the observed antenna brightness temperature in that direction.
Model Validation	The process of determining the degree to which a model is an accurate representation of the real-world from the perspective of the intended uses of the model. [Ref.: DoDD 5000.59-DoD Modeling and Simulation Management]
Model Verification	The process of determining that a model implementation accurately represents the developer's conceptual description and specifications. [Ref.: DoDD 5000.59-DoD Modeling and Simulation Management]

7.2 Acronyms

Table 6. Acronyms

ACO	Atmospheric Correction over Ocean
ADCS	Advanced Data Collection System
ADS	Archive and Distribution Segment
AFB	Air Force Base
AFM	Airborne Fluxes and Meteorology Group
AFSCN	Air Force Satellite Control Network
AFWA	Air Force Weather Agency
AFWWS	Air Force Weather Weapon System
AGE	Aerospace Ground Equipment
AIAA	American Institute of Aeronautics and Astronautics
ANSI	American National Standards Institute
Ao	Operational Availability
AOS	Acquisition of Signal
ATMS	Advanced Technology Microwave Sounder
BIT	Built-in Test
BITE	Built-in Test Equipment
BMMC	Backup Mission Management Center
C2	Command and Control
C3S	Command, Control, and Communications Segment
CCSDS	Consultative Committee for Space Data Systems
CDA	Command and Data Acquisition
CDDIS	Crustal Dynamics Data Information System
CDR	Climate Data Records
CERES	Cloud and Earth Radiant Energy System
CGMS	Coordination Group for Meteorological Satellites
CI	Configured Item
CLASS	Comprehensive Large-Array data Stewardship System
CMIS	Conical Microwave Imager Sounder
CMOC	Cheyenne Mountain Operations Center
COMSAT	Communications Satellite

COMSEC	Communications Security
CONUS	Continental United States
COTS	Commercial Off the Shelf
CrIMSS	Cross-Track Infrared Microwave Sounding Suite
CrIS	Cross-Track Infrared Sounder
CSCI	Computer Software Configured Item
DCP	Data Collection Platforms
DES	Digital Encryption System
DFCB	Data Format Control Book
DHN	Data Handling Node
DMSP	Defense Meteorological Satellite Program
DOC	Department of Commerce
DoD	Department of Defense
DRR	Data Routing and Retrieval
EDR	Environmental Data Records
EELV	Evolved Expendable Launch Vehicle
EMC	Electromagnetic Compatibility
EMD	Engineering and Manufacturing Development
EOL	End of Life
EOS	Earth Observing System
ERBS	Earth Radiation Budget Suite
ESD	Electrostatic Discharge
EUMETSAT	European Organization for the Exploitation of Meteorological Satellites
EWR	Eastern and Western Ranges
FFRDC	Federally Funded Research and Development Center
FMH	Federal Meteorological Handbook
FNMOC	Fleet Numerical Meteorology and Oceanography Center
FOC	Full Operational Capability
FTS	Field Terminal Segment
FVS	Flight Vehicle Simulator
GFE	Government Furnished Equipment
GIID	General Instrument Interface Document
GN	NASA Ground Network
GPS	Global Positioning System
GPSOS	GPS Occultation Suite
GSE	Ground Support Equipment
GSFC	Goddard Space Flight Center
HIJACK	Data Conversion Software
HRD	High Rate Data
IAW	In Accordance With
ICD	Interface Control Document
IDPS	Interface Data Processor Segment
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IGS	International GPS Service
IJPS	Initial Joint Polar System
ILS	Integrated Logistics Support
IOC	Initial Operational Capability
IORD	Integrated Operational Requirements Document
IOT&E	Initial Operational Tests & Evaluation
IP	Intermediate Product
IPL	Integrated Priority List
IPO	Integrated Program Office
IRD	Interface Requirements Document

ISO	International Standards Organization
ITRF	International Terrestrial Reference Frame
ITU	International Telecommunications Union
JPS	Joint Polar System
JSC	Johnson Space Center
JTA	Joint Technical Architecture
km	kilometer
LEO&A	Launch, Early Orbit, & Anomaly Resolution
LOS	Loss of Signal
LRD	Low Rate Data
LSS	Launch Support Segment
LST	Local Solar Time
LUT	Look-Up Table or Local User Terminal
LV	Launch Vehicle
MDT	Mean Down Time
Metop	Meteorological Operational Program
MMC	Mission Management Center
MOU	Memorandum of Understanding
MSS	Mission System Simulator
MTBCF	Mean Time Between Critical Failures
MTBDE	Mean Time Between Downing Events
MTTRF	Mean Time to Restore Function
NA	Non-Applicable
NACSEM	NPOESS Acquisition Cost Estimating Model
NASA	National Aeronautics and Space Administration
NAVOCEANO	Naval Oceanographic Office
NCA	National Command Authority
NCEP	National Centers for Environmental Prediction
NDT	Nitrate-Depletion Temperature
NESDIS	National Environmental Satellite, Data, and Information Service
NOAA	National Oceanic and Atmospheric Administration
NORAD	North American Aerospace Defense Command
NPOESS	National Polar-orbiting Operational Environmental Satellite System
NPP	NPOESS Preparatory Program
NSA	National Security Agency
NTIA	National Telecommunications Information Agency
OC/C	Ocean Color/Chlorophyll
O&M	Operations and Maintenance
OMPS	Ozone Mapping and Profiling Suite
P3I	Potential Pre-planned Product Improvements
PHS&T	Packaging, Handling, Storage, and Transportation
PIP	Program Implementation Plan
PM&P	Parts, Materials, and Processes
PMT	Portable Mission Terminal
POD	Precise Orbit Determination
POES	Polar Orbiting Environmental Satellite
RDR	Raw Data Records
RPIE	Real Property Installed Equipment
RSR	Remote Sensing Reflectance
S&R	Search and Rescue
SARSAT	Search and Rescue Satellite Aided Tracking
SCA	Satellite Control Authority
SDC	Surface Data Collection
SDE	Selective Data Encryption

SDP	Software Development Plan
SDR	Sensor Data Records
SDS	Science Data Segment
SESS	Space Environmental Sensor Suite
SGI®	Silicon Graphics, Inc.
SI	International System of Units
SMD	Stored Mission Data
SN	NASA Space Network
SOC	Satellite Operations Center
SRD	Sensor Requirements Documents
SS	Space Segment
SST	Sea Surface Temperature
STDN	Spaceflight Tracking and Data Network
SVE	Space Vehicle Equipment
TBD	To Be Determined
TBR	To Be Resolved
TBS	To Be Supplied
TDR	Temperature Data Records
TDRSS	Tracking and Data Relay Satellite System
TEMPEST	Telecommunications Electronics Material Protected from Emanating Spurious Transmissions
TOA	Top of the Atmosphere
TRD	Technical Requirements Document
TSIS	Total Solar Irradiance Sensor
USAF	United States Air Force
USB	Unified S-band
USG	United States Government
UTC	Universal Time Coordinated
VIIRS	Visible/Infrared Imager Radiometer Suite